5. Attack Identification, Recognition and Isolation via the Statistical Recognition Unit.

All the victim traffic that has passed the anti-spoofing authentication and was not stopped by the filter flows through the statistical unit. The statistical unit analyzes the traffic and identifies malicious sources (i.e., compromised sources), and provides operational rules for blocking the attack without disturbing innocent genuine traffic. The basic principle behind the unit's operation is that the pattern of traffic originating from a black-hat daemon drastically differs from the pattern generated by such a source during normal operation. In contrast, traffic patterns of "innocent" sources during an attack resemble their traffic at normal times. This principle is used to identify the attack sources and provide guidelines for their blockage by either the filter or the access lists of the routers. For example, the volume of a traffic from an attacking daemon, the distribution of packet sizes, port numbers, the distribution of the packets inter arrival times, and the ratio of inbound and outbound traffic are all parameters that may indicate that a source (client) is an attacking daemon.

The statistical unit has two major tasks:

1. Learning the traffic patterns during normal operation, i.e., when no attack is being mounted. These patterns are used while defending a victim during an attack to compare with the actual traffic in order to distinguish the malicious traffic from genuine traffic. We consider three possible ways in which this learning can be done: (1) Using the routers NetFlow data, (2) Analyzing the server logs at the victim server, and (3) Analyzing the potential victim traffic at the guard by having the traffic diverted to the guard from time to time for randomly sampling it.
2. Monitoring the victim traffic during an attack to identify and isolate the malicious traffic from the good genuine traffic. The identity of the attacking host is then given to the filter or the neighboring routers that would then drop any packet arriving from that host.

5.1 Network flows and traffic classification

The basic element studied by the statistical unit is a flow. Each flow is a sequence of packets belonging to the same connection. In the most general way a flow is identified by the following parameters: Source IP address, Source port, Destination port, Protocol type, time of day and day of week of connection creation. The destination IP address is implied since we collect all the information per destination address. For each such flow the traffic volume is registered.

Keeping all of the above information is infeasible since it requires an unacceptable amount of memory. However we employ learning methods to study the basic characteristics of the traffic destined to each destination and keep these key parameters succinctly, in an efficient way. Essentially the learning method studies the typical behavior of groups of users that interact with the destination. For example, a typical web site is accessed either by individual users sitting behind a host (pc), by a group of users sharing one multi user time sharing host, or by a group of users sitting behind a proxy. For each such group its typical behavior is studied. Other types of

users are possible such as web crawlers (for search engines) and monitoring servers such as keynote (www.keynote.com). Furthermore, for the largest group, i.e. the group of individual users, their identities (IP address) is not kept. Each source which is not included in the other groups is assumed to be an individual source. On the other hand, for the group of proxy machines that access the destination, the individual IP address of each is kept in a trie like data-structure. For other groups of users only their IP address may be needed since their traffic would be blocked from the beginning during an attack. Henceforth, the rest of this section considers types of users and the characteristic of flows originating from such users.

The basic parameters characterizing each user group are:

1. **Traffic volume distribution**: These include the **mean** and **variance** of the traffic such a user generates.
2. **Port numbers distribution**: Source port number distribution, and destination port number distribution.
3. **Periodicity:** Sources will be examined for the periodicity of their requests. It is likely that malicious sources act in a relatively periodic manner, while innocent sources act not in a regular manner.
4. **Packet Properties:** The distribution of packet sizes.

5.2 Learning Traffic Characteristics

There are three possible ways, that we consider, to learn and analyze the traffic characteristics of a particular target:

1. Sampling a **fraction** α of the **packets** $(0<\alpha\leq 1)$ traversing the lines on route to the target and then classifying the packets according to the flow id and time of day and day of week.

   Notice that setting α=1 requires the unit to process every packet and thus imposes high load on it while providing the best statistical measure. Lower values of α reduce the load posed on the unit while potentially somewhat degrading the statistical measure. The fraction α, therefore, will be a parameter that will be set so that enough statistical knowledge can be gained without over-loading the system.

2. Utilizing **server logs** collected by the defended target. These typically contain information about the activity being applied on the target. For example, WEB sites, which are likely to form the main body of potential targets, keep logs that record all the document requests sent to the site (including their source address, time of the day and other parameters). Processing of these logs by ZZZ yields a very accurate measure of the statistics of network flow volumes (measured in packets per second, as in a) above). The potential draw backs of this method are first that being collected at the target it is not immediately clear which information is relevant to which guarding point, and second, the pattern seen by the target may be slightly different from the pattern seen at the network boarder. However neither is a real problem and the first one may be a feature, since network routes change and the traffic may enter the network from a different point in any event.

3. Analyzing netflow data collected from the appropriated routers. This option requires the backbone provider to enable netflow and process it with our learning applications. This method has some limitation but none seems prohibitory. The limitations are that netflow aggregates information for each flow in intervals of a few minutes (typically 5 minutes intervals), and in this intervals it does not maintain the sizes of individual packets. Rather, it counts the total number of packets and bytes passed in this interval for each flow.

5.3 Traffic Monitoring and Analysis at Attack Time

In attack time while the guard machine defends a victim it monitors the victim traffic, classifies its traffic (incoming and outgoing) and compares the traffic to the normal traffic in order to detect the malicious traffic. Notice that during an attack information is collected only on the current flows. The information about well behaving flows is not kept more than small number of minutes.

1. **Online traffic volume collection at attack time:** This module collects the statistics of the traffic destined to the target(s) in attack time. Notice that in this sense, its measures are similar to the measures collected in approach 1i above. The classification of the traffic, in general, is similar to that conducted in the learning phase but may be controlled/guided by external intervention. Such intervention is enacted if some additional knowledge on the attack type is gained from other sources (e.g., human-aided identification) and can be utilized by the unit.
2. **Attack Analysis:** Is conducted in attack time and is responsible to compare the statistical data learned with the current traffic volume and generate rules for traffic blockage. The output of this unit, in general, will consist of a list of items for each of which three parameters will be provided:
   a. **Network flow**, identified by a combination of source IP address (can be prefixed), destination IP address, destination port number, protocol type (one may consider blockage that disregards port numbers, i.e., all the traffic originating from a compromised IP address, be it a proxy or a host).
   b. **Duration,** identifying the duration for which that class will be blocked.

   The analysis is based on the statistical parameters of the data and aims at keeping the target traffic at normal loads by blocking the most "suspicious" traffic streams. Blocking rules are based on maximizing the likelihood of blocking malicious traffic while minimizing the likelihood of blocking innocent traffic.

5.3.1 Statistical Recognition of Data "Innocence"

NETGUARDS uses two major properties of network flows to identify malicious traffic: a) Traffic pattern, and b) Traffic volume. Below we describe the recognition approaches based on these factors.

5.3.1.1 Recognition of Traffic Pattern

Several aspects of traffic pattern are examined:

1) **Source "IP geography" proximity**: Sources will be classified into classes that resemble the "IP geography", that is IP addresses that reside on neighboring networks (using IP address prefix) are classified in the same class. A class that generates a relatively large volume of requests is suspected as being malicious. Notice, that such "malicious classes" are likely to form if the attacker planted a collection of daemons in the same network, and this network does not use a proxy.

2) **Periodicity:** Sources will be examined for the periodicity of their requests. It is likely that malicious sources act in a relatively periodic manner, while innocent sources act in more irregular pattern.

3) **Packet Properties:** Sources will be examined for repetitive properties of their packets. For example the distribution of packet size. It is likely that malicious sources generate packets of identical properties (e.g. – all packets of same size) while innocent sources will generate packets of more random nature. Other properties include port number distributions.

5.3.1.2 Recognition of Traffic Volume

Traffic volume recognition is used to identify malicious sources that transmit **large volumes** of data which **significantly differ** from their normal volume. Specifically, we classify Internet data sources to *small sources* and *large sources.* The former relates to individual IP addresses whose traffic volume is normally tiny. The latter relates to Proxy traffic or Spider (Crawler) traffic[1] whose volume is drastically higher.

ZZZ keeps individual volume parameters for each of the large sources. Individual parameters are not kept for the small sources; rather a single fixed small number (related to their mean volume averaged over all these sources) will be recorded. At attack time the traffic volumes of individual flows will be measured and compared to their recorded volume. Flows whose volume drastically differs (upwards) from their recorded measure are marked as being malicious.

The mathematical formulation of this procedure is as follows: Given are K classes of flows, indexed *1, 2, ..., K,* and characterized by the mean (μ_i) and the variance (σ_i) of their learned volume, and by their current volume (X_i). We would like to identify the classes with the largest deviation from their corresponding expected volume. Let $Y_i = (X_i - \mu_i)/\sigma_i$. We will sort the classes

[1] The traffic volume resulting from a Spider access is normally higher than that of a single human user, especially on large WEB sites. The reason is that a spider scans the whole site, leading to hundred or thousands of requests while a human client requests tens of pages or less, on average.

by the value of Y_i and recommend blocking the classes with the largest values of Y_i.

5.3.2 Time accumulating traffic volume recognition and "controlled" denial of service

It is important to recognize that the effectiveness of volume recognition increases with the time duration along which it is implemented. This is correct since the variance of total data volume generated by a source during a period of duration *T* decreases in *T*. For example, it is expected that the average amount of traffic generated by a small source during a period of 1 hour will be *very small.* However, at certain epochs, it is expected that the average amount of traffic generated by the same source during a period of 1 minute can be rather large. (up to 60 times larger than that of the 1 hour average).

For this reason ZZZ will implement the following unique recognition and traffic screening mechanism. For source *i*, let $S_i(t)$ denote the amount traffic generated by the source during the interval *(0,t)* (where we assume that the attack starts at time 0). We then set at time *t:* $X_i(t) = S_i(t)/t$ and apply the above screening mechanism.

This mechanism has the following properties:

1. For a small value of *t* (that is, at the first few minutes of the attack) a sophisticated attacker might cause significant number of innocent users denial of service. This is due to the fact that the attacker may inflict a load that resembles that of an innocent client, and thus the attacker is not distinguishable from the innocent client. At this stage, ZZZ may block some innocent clients and some attackers. Using this action, for a short period of time, some innocent clients may be denied of service but ZZZ protects the site from going down.
2. As *t* increases more and more malicious sources are identified and blocked and fewer innocent sources are blocked. This is since the malicious sources have posed large amount of accumulated load. Thus as time progresses less and less innocent clients are denied service. In fact, after relatively short period all malicious sources will be denied service while the innocent sources will receive full regular service.

Example: Consider the traffic volume generated on the web site of the Nagano server (Feb 98). It had 11,665,713 requests made over a period of 24 hours by 59,582 clients. Assuming uniform distribution of clients over the day,

this implies about 2500 clients per hour and 500 clients per 12-minute interval. An attacker who uses 500 sophisticated daemons (which imitate a normal client) will look innocent at the first 12 minutes interval. At this period ZZZ will block 50% of the innocent clients and 50% of the daemons. However, after 24 minutes the daemons will generate significantly more traffic than an innocent client and thus almost all of the traffic blocked will be that of malicious daemons.